6.0 INPUT DATA

6.1 General Considerations

6.1.1 Techniques

6.1.1.1 Control file or terminal I/O startup

The input/output (I/O) file information is provided to the code from an input control file or the terminal. The default control file name is *fehmn.files*. If a control file with the default name is present in the directory from which the code is being executed, no terminal input is required. If the default control file is not present, input prompts are written to the screen preceeded by a short description of the I/O files used by FEHM. The descriptions of the I/O files are elaborated on in Section 5.0. The initial prompt asks for the name of a control file. If a control file name is entered for that prompt, no further terminal input is required. If a control file is not used, the user is then prompted for I/O file names, the tty output flag, and user subroutine number. When the input file name is entered from the terminal, the user has the option of letting the code generate the names for the remainder of the auxiliary files using the input-file-name prefix. The form of the input file name is *filen* or *filen.**, where "*filen*" is the prefix used by the code to name the auxiliary files and ".*" represents an arbitrary file extension.

6.1.1.2 Macro control structure

The finite-element heat- and mass-transfer code (FEHM) contains a macro control structure for data input that offers added flexibility to the input process. The macro command structure makes use of a set of control statements recognized by the input module of the program. When a macro control statement is encountered in an input file, a certain set of data with a prescribed format is expected and read from the input file. In this way, the input is divided into separate, unordered blocks of data. The input file is therefore a collection of macro control statements, each followed by its associated data block. Blocks of data can be entered in any order, and any blocks unnecessary to a particular problem need not be entered. The macro control statements must appear in the first four columns of a line. The other entries are free format, which adds flexibility but requires that values be entered for all input variables (no assumed null values).

As an aid to the user, the capabilities of FEHM summarized in Table I refer to applicable macro commands. Table V lists the macro control statements with a brief description of the data associated with each. A more detailed description of each macro control statement and its associated input are found in Section 6.2. Macro control statements may be called more than once, if for example, the user wishes to reset some property values after defining alternate zones. Some statements are required, as indicated in Table V; the others are optional.

Table V. Macro control statements for FEHM

Control statement	Description
adif	Air-water vapor diffusion
airwater	Isothermal air-water input
alti	Alternate input
bous	Boussinesq-type approximation
cap	No longer used, see macro rlp
cond	Thermal-conductivity data **(required for nonisothermal problem)**
cont	Contour plot data
coor	Node-coordinate data **(required)**
ctrl	Program-control parameters **(required)**
dof	[Not implemented]
dpdp	Double-porosity/double-permeability model input
dual	Input for dual-porosity solution
elem	Element-node data **(required)**
eos	Equation-of-state data
finv	Finite-volume flow coefficients
flow	Flow data
flo2	Alternate format for flow data
flxo	Flux printout
hflx	Heat-flux input
ice	Ice-phase calculations (untested)
init	Initial value data **(required if macro pres or restart file is not used)**
iter	Iteration parameters
itup	Iterations used with upwinding
iupk	Upwind transmissibility including intrinsic permeability
ivfc	Enables volume-factor calculations
mdnode	Enables extra connections to be made to nodes
ngas	Noncondensible-gas (air) data
nod2	Node numbers for output and time histories, and alternate nodes for terminal output
node	Node numbers for output and time histories

Table V. Macro control statements for FEHM (continued)

Control statement	Description
num	[Not implemented]
perm	Permeability input **(required)**
ppor	Pressure- and temperature-dependent porosity and permeability
pres	Initial pressure, temperature, and saturation data, boundary-conditions specification **(required if macro init or restart file is not used)**
ptrk	Particle-tracking simulation input
renm	Renumbers nodes
rflx	Radiation-flux input
rlp	Relative-permeability input **(required for 2-phase problem, otherwise optional)**
rock	Rock-density, specific-heat, and porosity input **(required)**
rxn	Chemical-reaction-rate model input
sol	Solver specifications **(required)**
solv	[Not implemented]
stea	Steady-state solution generated for initial variable field
stop	Signals the end of input **(required)**
strs	Initiates stress solution [Not implemented in this version of FEHM]
text	Text input to be written to output file
thic	Variable-thickness input for two-dimensional problems
time	Time-step and time-of-simulation data **(required)**
trac	Solute-simulation input
user	User subroutine call
vcon	Variable-thermal-conductivity input
velo	Velocity printout
wlbr	Wellbore-style input
zone	Geometric definition of grid for input-parameter assignment

Comments may be entered in the input file by beginning a line with a "#" symbol (the "#" symbol must be found in the first column of the line). Comments may precede or follow macro blocks but may not be found within a block.

Optional input files may be used by substituting a keyword and file name in the main input file (described in detail in

Section 6.2.2). The normal macro input is then entered in the auxiliary file.

Many input parameters, such as porosity or permeability, vary throughout the grid and need to have different values assigned at different nodes. This task is accomplished in two ways. The first uses a nodal loop-type definition (which is the default):

JA, **JB**, **JC**, PROP1, PROP2 ... ,

where

JA - first node to be assigned with the properties PROP1, PROP2 ... ,

JB - last node to be assigned with the properties PROP1, PROP2 ... ,

JC - loop increment for assigning properties PROP1, PROP2 ... , and

PROP1, PROP2 - property values to be assigned to the indicated nodes.

In the input blocks using this structure, one or more properties are manually entered in the above structure. When a blank line is entered, that input block is terminated and the code proceeds to the next group or control statement. (Note that blank input lines are shaded in the examples shown in Section 6.2.). The nodal definition above is useful in simple geometries for which the node numbers are easily found. Boundary nodes often come at regular node intervals, and the increment counter **JC** can be adjusted so the boundary conditions are easily entered. To set the same property values at every node, the user may set **JA** and **JC** to 1 and **JB** to the total number of nodes or, alternatively, set **JA** = 1 and **JB** = **JC** = 0.

For dual-porosity problems, which have three sets of parameter values at any nodal position, nodes 1 to N [where N is the total number of nodes in the grid (see macro **coor**)] represent the fracture nodes, nodes N + 1 to 2N are generated for the second set of nodes, the first matrix material, and nodes 2N + 1 to 3N for the third set of nodes, the second matrix material. For double-porosity/double-permeability problems, which have two sets of parameter values at any nodal position, nodes 1 to N represent the fracture nodes and nodes N + 1 to 2N are generated for the matrix material.

For more complicated geometries, such as 3-D grids, the node numbers are often difficult to determine. Here a geometric description is preferred. To enable the geometric description, the **zone** control statement (page 69) is used in the input file before the other property macro statements occur. The input macro **zone** requires the specification of the coordinates of 4-node parallelograms for 2-D problems or 8-node polyhedrons in 3-D. In one usage of the control statement **zone**, all the nodes are placed in geometric zones and assigned an identifying number. This number is then addressed in the property input macro commands by specifying a **JA** < 0 in the definition of the loop parameters given above. For example, if **JA** = -1, the properties defined on the input line would be assigned to the nodes defined as belonging to geometric Zone 1 (**JB** and **JC** must be input but are ignored in this case). The control statement **zone** may be called multiple times to redefine geometric groupings for subsequent input. The previous zone definitions are not retained between calls. Up to 100 zones may be defined. For dual-porosity problems, which have three sets of parameter values at any nodal position, Zone 100 + I is the zone number for the second set of nodes defined by Zone I, and Zone 200 + I is the zone number for the third set of nodes defined by Zone I. For double-porosity/double-permeability problems, which have two sets of parameter values at any nodal position, Zone 100 + I is the zone number for the second set of nodes defined by Zone I.

6.1.2 Consecutive cases

The program retains no input data between cases. The values of all variables are reinitialized with each run, either from the input files or a restart file when used.

6.1.3 Defaults

Default values are set during the initialization process if overriding input is not provided by the user.

6.2 Individual Input Records or Parameters

Other than the control file or terminal I/O, the main user input is provided using macro control statements in the input file, geometry data file, or zone data file. Data provided in the input files are entered in free format with the exception of the macro control statements, which must appear in the first four columns of a line. Data values may be separated with spaces, commas, or tabs. The primary input file differs from the others in that it begins with a title line (80 characters maximum) followed by input in the form of the macro commands. Each file containing macro commands should be terminated with the **stop** control statement. In the examples provided in the following subsections, blank input lines are depicted with shading.

6.2.1 Control file or terminal I/O input

The parameters enumerated below are entered in order, one per line, in the control file (excluding the control file name [nmfile(1)]) or as prompted for during terminal input. If there is a control file with the name fehmn.files in your local space, FEHM will execute using that control file, and there will be no prompts. If another name is used for the control file, it can be entered at the first prompt.

A blank line can be entered in the control file for any auxiliary files not required, for the "none" option for tty output, and for the "0" option for the user subroutine number. The code will always write an input check file and a restart file, so if names are not provided by the user the defaults will be used. If an ouput file name is not specified, the generalized output is written to the terminal.

Input variable	Format	Opt/Req	Default	Description
nmfil(1)	character*100	Opt	fehmn.files	Control file name (this line is not included in the control file).
nmfil(2)	character*100	Req	fehmn.dat	Main input file name.
nmfil(3)	character*100	Opt	not used	Geometry-data input file name.
nmfil(4)	character*100	Opt	not used	Zone-data input file name.
nmfil(5)	character*100	Opt	terminal	Main output file name.
nmfil(6)	character*100	Opt	not used	Restart input file name.
nmfil(7)	character*100	Opt	fehmn.fin	Restart output file name.
nmfil(8)	character*100	Opt	not used	Simulation-history output file name.
nmfil(9)	character*100	Opt	not used	Solute-history output file name.
nmfil(10)	character*100	Opt	not used	Contour-plot output file name (required if using avs option in cont macro).
nmfil(11)	character*100	Opt	not used	Dual-porosity or double-porosity/double-permeability contour-plot output file name.
nmfil(12)	character*100	Opt	not used	Coefficient-storage output file name.
nmfil(13)	character*100	Opt	fehmn.chk	Input-check output file name.
tty_flag	character*4	Opt	none	Terminal output flag: all, some, none.
usub_num	integer	Opt	0	User subroutine call number.

The following are examples of input control files.

```
tape5.dat
tape5.dat
tape5.dat
tape5.out


tape5.his
tape5.trc
tape5.con


tape5.chk
some
0
```

```
/groupdir/c14-3
/groupdir/grid-402
/groupdir/c14-3
c14-3.out
/groupdir/c14-3.ini
c14-3.fin
c14-3.his
c14-3.trc
c14-3.con
c14-3.dp
c14-3.stor
c14-3.chk
none
0
```

6.2.2 Optional input files

The data for any of the FEHM macros (with the exception of **coor** and **elem**) may be entered in an alternate input file. To use this option, the keyword "file" must appear on the input line immediately following the control statement (macro name). The line immediately following this keyword will contain the name of the alternate input file. The contents of the alternate input file consist of the regular full macro description: the macro name followed by the data. The entries in the optional input file may be preceded or followed by comments using the “#” designator (see discussion on page 18). As with regular macro input, comments may not be embedded within the data block.

Group 1 - LOCKEYWORD

Group 2 - LOCFILENAME

Input variable	Format	Description
LOCKEYWORD	character*4	Keyword “file” to designate an auxiliary input file is used.
LOCFILENAME	character*100	Name of the optional data input file.

The following illustrate the use of an optional file and its contents.

```
rock
file
rockfile
```

File “rockfile”:

```
# Auxiliary file used for rock macro input
rock
   1       140         1      2563.        1010.         0.3500

# End of rock macro input
```

6.2.3 Control statement adif (optional)

Air-water vapor diffusion.

Group 1- TORT

Input variable	Format	Description
TORT	real	Tortuosity for air-water vapor diffusion.

6.2.4 Control statement airwater (optional)

Isothermal air-water two-phase simulation.

Several macros are affected if the air module is enabled. These are:

pres - Because the air-water formulation is 2-phase at all times, care should be taken to insure that IEOSD is always specified to be 2. Likewise, saturations (not temperatures) are used.

init - This macro should not be used because the saturation values cannot be specified.

flow - A variety of different flow and boundary values are input with this macro when the macro **airwater** is also used. See description of control statement **flow** (page 34).

Group 1 - ICO2D

Group 2 - TREF, PREF

Input variable	Format	Description
ICO2D	integer	Determines the type of air module used. ICO2D = 1, 1-degree-of-freedom solution to the saturated-unsaturated problem is produced. This formulation is similar to the Richard's Equation. ICO2D = 2, 1-degree-of-freedom solution is obtained assuming only gas flow with no liquid present. ICO2D = 3, full 2-degrees-of-freedom solution. All other values are ignored. The default is [3].
TREF	real	Reference temperature for properties (°C).
PREF	real	Reference pressure for properties (MPa).

The following is an example of **airwater**

```
airwater
   3
  20.          0.1
```

6.2.5 Control statement alti (optional)

Alternate element and coordinate input. Not supported in this version.

Group 1 - CC, N

Group 2 - INFL

Input variable	Format	Description
CC	character*4	Input file type (ment: mentat mesh generator; ptrn: patran mesh generator).
N	integer	Number of nodes in the grid.
INFL	character*100	Name of alternate element- and coordinate-data input file.

6.2.6 Control statement bous (optional)

Boussinesq-type approximation. The flow terms have constant properties except for the relative permeability. Only enabled for the air-water isothermal physics package (see macro **airwater**, page 22).

Group 1 - ICONS

Input variable	Format	Description
ICONS	integer	Any nonzero integer will enable this macro.

6.2.7 **Control statement cap** (no longer used, see macro **rlp**, page 50)

6.2.8 **Control statement cond** (required for nonisothermal problem)

Assign thermal conductivities of the rock.

Group 1 - JA, JB, JC, THXD, THYD, THZD (JA, JB, JC - defined on page 19)

Input variable	Format	Default	Description
THXD	real	1.e-30	Thermal conductivity in the x-direction $\left(\frac{W}{m \cdot K}\right)$.
THYD	real	1.e-30	Thermal conductivityin the y-direction $\left(\frac{W}{m \cdot K}\right)$.
THZD	real	1.e-30	Thermal conductivity in the z-direction$\left(\frac{W}{m \cdot K}\right)$.

The following is an example of **cond**

```
cond
   1       140        1         1.00e-00        1.00e-00        0.00e-00
```

6.2.9 **Control statement cont** (optional)

Contour-data output format, output time-step intervals, and time intervals.

Group 1 - NCNTR, CONTIM

An alternative form of input for macro **cont** is possible. This is:

Group 1 - ALTC, NCNTR, CONTIM

Group 2 - CHDUM (only input if ALTC is 'avs')

FEHM will automatically distinguish between the alternative input formats. When keywords are used, they must be entered starting in the first column. The contour data will be output whenever either of the interval criteria are satisfied.

For AVS output, if the *material* keyword is selected, the following material property values will be written for each node: permeability in the x-, y-, and z-directions, thermal conductivity in the x-, y-, and z-directions, porosity, rock specific heat, capillary pressure, relative-permeability model being used, and capillary-pressure model being used. If *vapor* and/or *liquid* are selected, *pressure* or *velocity* must also be defined (otherwise, no data for these values will be written). *velocity* will result in vector values; *pressure* values will be scalar. If *concentration* is selected, values will be output only if nspeci is defined for tracer solutions. See the control statement **trac** for a description of nspeci for solutes.

Input variable	Format	Description
ALTC	character*4	Keyword specifying the type of contour output wanted (avs, fehm, free, ment, ptrn): 'avs' produces contour plot files compatible with the AVS postprocessor. 'fehm' produces a binary output file. The same contour plot file is produced using the first form of Group1 input. 'free' produces a free-format contour plot file. 'ment' produces a contour plot file compatible with the MENTAT postprocessor. 'ptrn' produces a contour plot file compatible with the PATRAN postprocessor.
NCNTR	integer	*time-step* interval for contour plots (number of time steps). Output contour information each NCNTR time steps.
CONTIM	real	*Time* interval for contour plots (days). In addition to output each NCNTR time steps, output contour information each CONTIM days.
CHDUM	character*72	Keyword specifying type of AVS contour-plot data files to be created in AVS UCD format, either formatted (ASCII) or unformatted (binary). Keywords are entered one per line and terminated with 'endavs'. Valid keywords (case insensitive) are: (m)aterial - output contour values for material properites. (l)iquid - output contour values for liquid phase. (va)por - output contour values for vapor phase. (ve)locity - output velocity values. (dp)dp - output contour values for dual-permeability nodes. (p)ressure - output pressure values. (t)emperature - output temperature values. (s)aturation - output saturation values. (c)oncentration - output solute-concentration values. (f)ormatted - output data in ASCII format. (u)nformatted - output data in binary format. (e)ndavs - last keyword entered. If a format keyword is not entered, the default is 'formatted'. The default for data keywords is "off." The letters given in ( ) are sufficient to identfy the keyword.

The following are examples of **cont**. For the first example, FEHM binary-format contour output files will be written every 100 time steps and for each 1.e20 days. The second example invokes AVS contour output. AVS UCD binary files will be written for every 100 time steps and 1.e20 days. The resulting files will include a log file, geometry file, plus header and data files for the following: material properties, solute concentrations, liquid velocities, pressures, and temperatures.

```
cont
 100            1.e20
```

```
cont
avs                  100             1.e20
liquid
velocity
con
pressure
temp
mat
unformatted
end
```

6.2.10 Control statement coor (required)

Node coordinate data. These data are usually created by a mesh-generation program, then cut and copied into the input file or a separate geometry-data input file. The mesh must be a right-handed coordinate system.

Group 1 - N

Group 2 - MB, CORD1, CORD2, CORD3

To end the control section a blank line is entered.

Input variable	Format	Description
N	integer	Number of nodes in the grid.
MB	integer	Node number. If MB < 0, then the difference between the absolute value of MB and the previously read absolute value of MB is used to generate intermediate values by interpolation.
CORD1	real	X-coordinate of node MB (m).
CORD2	real	Y-coordinate of node MB (m).
CORD3	real	Z-coordinate of node MB (m).

The following is an example of **coor**

```
coor
 140
   1        0.00000     200.00000       0.00000
   2       12.50000     200.00000       0.00000
   .           .            .              .
   .           .            .              .
   .           .            .              .
  10      212.50000     200.00000       0.00000
   .           .            .              .
   .           .            .              .
   .           .            .              .
 140      300.00000       0.00000       0.00000
```

6.2.11 Control statement ctrl (required)

Assigns various control parameters needed for equation-solver and matrix-solver routines.

Group 1 - MAXIT, EPM, NORTH

Group 2 - JA, JB, JC, IGAUS (JA, JB, JC - defined on page 19)

Group 3 - AS, GRAV, UPWGT

Group 4 - IAMM, AIAA, DAYMIN, DAYMAX

Group 5 - ICNL, LDA

Input variable	Format	Default	Description
MAXIT	integer		Maximum number of iterations allowed in either the overall Newton cycle or the inner cycle to solve for the corrections at each iteration. If MAXIT < 0, then the maximum number of iterations is ABS(MAXIT), but the minimum number of iterations is set to 2. [10]
EPM	real		Tolerance for Newton cycle (nonlinear equation tolerance). [1.e-5]
NORTH	integer		Number of orthogonalizations in the linear-equation solver. [8]
IGAUS	integer	1	The order of partial Gauss elimination (1 or 2 is recommended). Larger values increase memory use but may be necessary for convergence.
AS	real		Implicitness factor. [1] AS ≤ 1, use standard pure implicit formulation. AS > 1, use second-order implicit method.

GRAV	integer		Direction of gravity.
			GRAV = 0, no gravity is used.
			GRAV = 1, x-direction.
			GRAV = 2, y-direction.
			GRAV = 3, z-direction.
			A value for gravity of 9.81 m/s^2 is used in the code when GRAV ≠ 0. If GRAV > 3, GRAV is set equal to 3.

UPWGT	real		Value of upstream weighting (0.5 ≤ UPWGT ≤ 1.0). If UPWGT < 0.5, UPWGT is set to 0.5. If UPWGT > 1.0, UPWGT is set to 1.0.
IAMM	integer		Maximum number of iterations for which the code will multiply the time-step size. If this number of time steps is exceeded at any time, the time step will not be increased for the next time. [7-10]
AIAA	real	1	Time-step multiplier. [1.2-2.0]
DAYMIN	real		Minimum time-step size (days).
DAYMAX	real		Maximum time-step size (days).

Input variable	Format	Default	Description

ICNL	integer		Parameter that specifies the geometry.
			ICNL = 0, three-dimensional.
			ICNL = 1, x-y plane.
			ICNL = 2, x-z plane.
			ICNL = 3, y-z plane.
			ICNL = 4, x-y radial plane, (radius is x).
			ICNL = 5, x-z radial plane, (radius is x).
			ICNL = 6, y-z radial plane, (radius is y).

LDA	integer	0	Parameter that specifies the external storage of geometric coefficients. LDA = +1, element coefficients are read from file *filen.stor,* and no coefficients are calculated in the code. LDA = 0, element coefficients are calculated in the code and not saved. LDA = -1, element coefficients are calculated in the code and saved on file *filen.stor.* It should be noted that if the coefficients are read from a file (LDA = 1), then the macro **finv** is ignored as well as information read from macros **elem** and **coor**.

The following is an example of **ctrl**

```
ctrl
   40          1.e-7       8
    1      140             1           1

     1.0         0.0       1.0
   40            1.2       0.1         60.0
    1          0
```

6.2.12 Control statement dof (Not implemented)

6.2.13 Control statement dpdp (optional)

Double-porosity/double-permeability formulation. There are two sets of parameter values at any nodal position for which property values must be defined. Nodes 1 to N (see macro **coor**, page 26, for definition of N) represent the fracture nodes and nodes N + 1 to 2N, the matrix material. When zones are used with the **dpdp** macro, additional zones are automatically generated. See instructions for the macro **zone** for a more detailed description. The **dpdp** parameters are only defined for the first N nodes.

Group 1 - IDPDP

Group 2 - JA, JB, JC, VOLFD1 (JA, JB, JC - defined on page 19)

Group 3 - JA, JB, JC, APUV1 (JA, JB, JC - defined on page 19)

The volume fraction VOLFD1 is related to the total volume by

$$VOLFD1 + VOLFD2 = 1.0 ,$$

where VOLFD2 is the volume fraction of the matrix node. If permeability

Input variable	Format	Default	Description
IDPDP	integer		Solution descriptor for double-porosity/double-permeability solution. IDPDP = 0, information is read but not used. IDPDP ≠ 0, **dpdp** solution is implemented.
VOLFD1	real	1.	Volume fraction for fracture node.
APUV1	real	10.	Length scale for matrix nodes (m).

model IRLP = 4 is selected in control statement **rlp**, VOLFD1 is calculated from RP15 (fracture porosity) in that control statement.

The following is an example of **dpdp**

```
dpdp
   1
   1        140        1        0.005

   1        140        1        0.10

```

6.2.14 Control statement dual (optional)

Dual-porosity formulation. There are three sets of parameter values at any nodal position for which property values must be defined. Nodes 1 to N (see macro **coor**, page 26, for definition of N) represent the fracture nodes, nodes N + 1 to 2N, the first matrix material, and nodes 2N + 1 to 3N, the second matrix material. When zones are used with the **dual** macro, additional zones are automatically generated (see instructions for the macro **zone**, page 69, for a more detailed description). The **dual** parameters are only defined for the first N nodes.

Group 1 - IDUALP

Group 2 - JA, JB, JC, VOLFD1 (JA, JB, JC - defined on page 19)

Group 3 - JA, JB, JC, VOLFD2 (JA, JB, JC - defined on page 19)

Group 4 - JA, JB, JC, APUVD (JA, JB, JC - defined on page 19)

Input variable	Format	Default	Description
IDUALP	integer		Solution descriptor for dual-porosity solution. IDUALP = 0, information is read but not used. IDUALP ≠ 0, dual-porosity solution is implemented.
VOLFD1	real	0.001	Volume fraction for fracture portion of the continuum.
VOLFD2	real	0.5	Volume fraction for the first matrix portion of the continuum.
APUVD	real	5.	Length scale for the matrix nodes (m).

The volume fractions VOLFD1 and VOLFD2 are related to the total volume by

$$VOLFD1 + VOLFD2 + VOLFD3 = 1.0 ,$$

where VOLFD3 is the volume fraction of the second matrix node. If permeability model IRLP = 4 is selected in control statement **rlp**, VOLFD1 is calculated from RP15 (fracture porosity) in that control statement.

The following is an example of **dual**

```
dual
   1
   1        140          1          0.006711409

   1        140          1          0.335570470

   1        140          1          0.10

```

6.2.15 Control statement elem (required).

Element connectivity data. These data are created by a mesh-generation program, then cut and copied into the input file or a separate geometry-data input file.

Group 1 - NS, NEI

Group 2 - MB, NELM (1), NELM (2), . . ., NELM (NS)

IF $NS < 0$, then ABS(NS) is interpreted as the number of nodes per element. $NS < 0$ signals the code to make rectangles (or bricks in three dimensions) a sum of triangles (or tetrahedrons). This approach provides more stability in nonlinear problems with a distorted mesh. Figure 2 shows available element types and the nodal numbering convention. To end the control section a blank line is entered.

Input variable	**Format**	**Description**
NS	integer	Number of nodes per element.
NEI	integer	Number of elements.
MB	integer	Element number. If MB < 0, then the difference between the absolute value of MB and the previous absolute value of MB is used to generate intermediate values by interpolation in the code.
NELM (1)	integer	First node of element MB.
NELM (2)	integer	Second node of element MB.
⋮	⋮	⋮
NELM (NS)	integer	Last node of element MB.

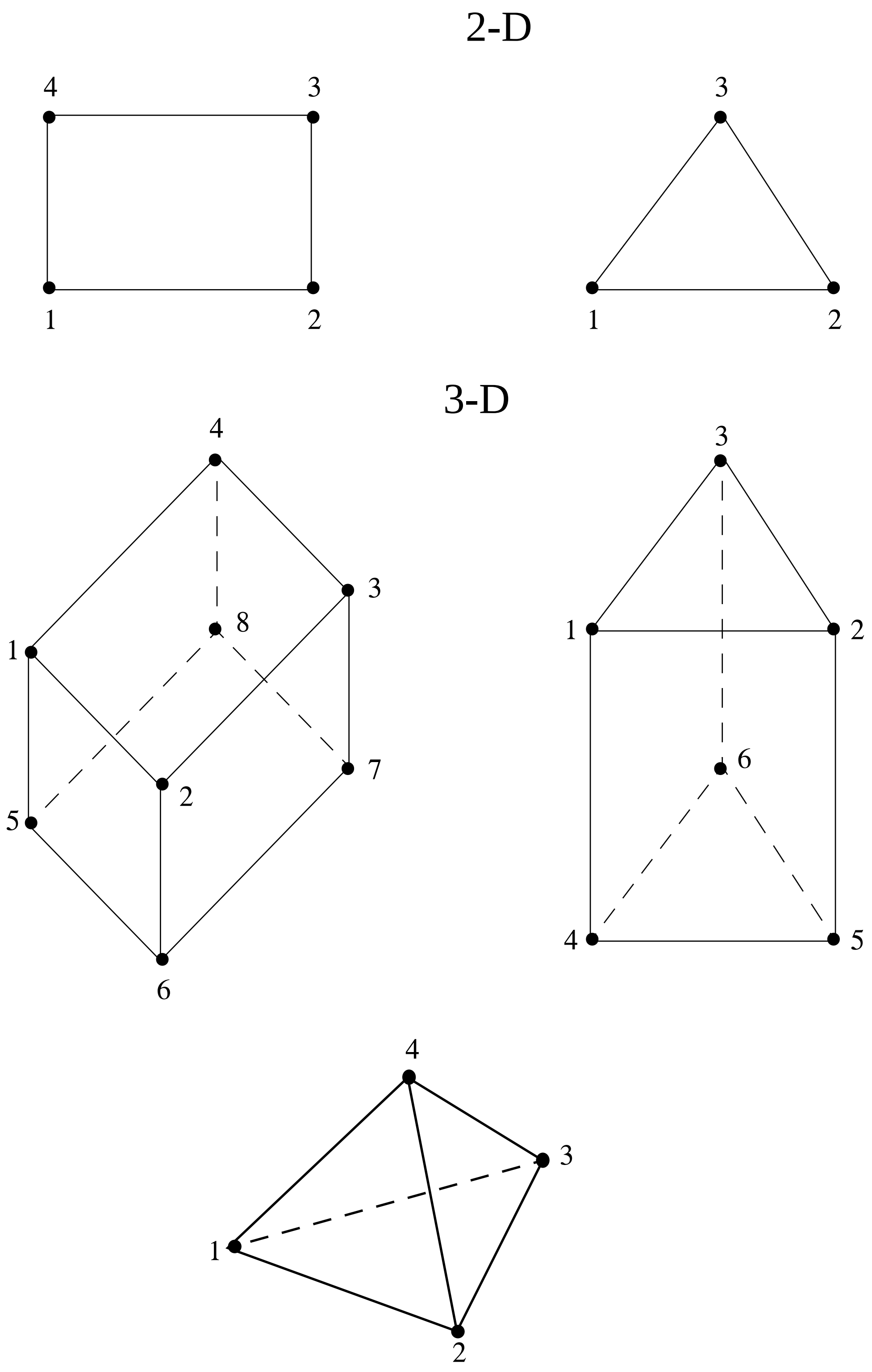

Figure 2. Elements available with FEHM in 2-D and 3-D problems showing nodal numbering convention.

The following is an example of **elem**

```
elem
   4      117
   1       15       16        2        1
   2       16       17        3        2
   .        .        .        .        .
   .        .        .        .        .
   .        .        .        .        .
  10       24       25       11       10
  11       25       26       12       11
  12       26       27       13       12
   .        .        .        .        .
   .        .        .        .        .
   .        .        .        .        .
 116      138      139      125      124
 117      139      140      126      125
```

6.2.16 Control statement eos (optional)

Equation of State. Provide the code with alternate thermodynamic properties for the liquid and/or vapor phases. (This is one way in which the code may be instructed to simulate nonisothermal, single-phase air. It may also be used to make comparisons between the code and analytical solutions that use different equations of state.)

Group 1 - IIEOSD, IPSAT, ITSAT

Group 2 - EWI, EW2, EW3, EW4, EW5, EW6, EW7, EW8, EW9, EW10, EW11

Group 3 - EVI, EV2, EV3, EV4, EV5, EV6, EV7, EV8, EV9, EV10, EV11

For the calculation of vapor density and its derivatives, the ideal gas law is used instead of a linear relationship. Thus, EV4 and EV5 are not used but are included so the format is the same as that for the liquid parameters in Group 2.

Input variable	**Format**	**Description**
IIEOSD	integer	Equation-of-state reference number. When IIEOSD = 1 or 2 are used, they refer to the high- and low-pressure data sets, respectively, in FEHM. For these values, the input in Group 2 and Group 3 will be ignored after it is entered. When any value other than 1 or 2 are used, the user-defined equation of state is used with Groups 2 and 3 for input.
IPSAT	integer	Parameter to set vapor pressure to zero. If IPSAT ≠ 0, the vapor pressure is set to zero, otherwise the vapor pressure is calculated in the code.
ITSAT	integer	Parameter to adjust the saturation temperature. If ITSAT < 0, the saturation temperature is set to -1000°C. If ITSAT > 0, the saturation temperature is set to 1000°C. If ITSAT = 0, the calculated value is used.
EW1	real	Liquid reference pressure (MPa).
EW2	real	Liquid reference temperature (°C).
EW3	real	Liquid reference density (kg/m^3).

Input variable	Format	Description
EW4	real	Derivative of liquid density with respect to pressure at reference conditions.
EW5	real	Derivative of liquid density with respect to temperature at reference conditions.
EW6	real	Liquid reference enthalpy (MJ/kg).
EW7	real	Derivative of liquid enthalpy with respect to pressure at reference conditions.
EW8	real	Derivative of liquid enthalpy with respect to temperature at reference conditions.
EW9	real	Liquid reference viscosity.
EW10	real	Derivative of liquid viscosity with respect to pressure at reference conditions.
EW11	real	Derivative of liquid viscosity with respect to temperature at reference conditions.
EV1	real	Vapor reference pressure (MPa).
EV2	real	Vapor reference temperature (°C).
EV3	real	Vapor reference density (kg/m^3).
EV4	real	Not used, included only to maintain a similar format to Group 2.
EV5	real	Not used, included only to maintain a similar format to Group 2.
EV6	real	Vapor reference enthalpy (MJ/kg).
EV7	real	Derivative of vapor enthalpy with respect to pressure at reference conditions.
EV8	real	Derivative of vapor enthalpy with respect to temperature at reference conditions.
EV9	real	Vapor reference viscosity.
EV10	real	Derivative of vapor viscosity with respect to pressure at reference conditions.
EV11	real	Derivative of vapor viscosity with respect to temperature at reference conditions.

6.2.17 Control statement finv (optional)

No input is associated with this macro. When invoked, the code will perform finite-volume calculations instead of finite-element calculations for flow terms—this may improve accuracy on nonorthogonal grid systems. Anisotropic properties (permeability, conductivity) are not supported with this macro. In this case, the values for permeability in the x-direction from control statement **perm** are used.

6.2.18 Control statement flow (required for flow problem)

Flow data. Source and sink parameters are input and may be used to apply boundary conditions. Note that the alternative definitions (isothermal conditions) apply when control statement **airwater** is used.

Group 1 -JA, JB, JC, SKD, EFLOW, AIPED (JA, JB, JC - defined on page 19)

If the porosity of the node is zero, then there is only a temperature solution, and the code forms a source proportional to the enthalpy difference. The source term is given by $Q = AIPED \cdot (E - EFLOW)$, where E is the in-place enthalpy and EFLOW is a specified enthalpy.

Input variable	Format	Default	Description
Nonisothermal			
SKD	real	0.	Heat and mass source strength (kg/s), heat only (MJ/s). Negative value indicates injection into the rock mass.
EFLOW	real	0.	Enthalpy of fluid injected (MJ/kg). If the fluid is flowing from the reservoir, then the in-place enthalpy is used. If EFLOW < 0, then ABS(EFLOW) is interpreted as a temperature (°C) and the enthalpy (assuming water only) calculated accordingly. In heat-only problems with EFLOW < 0, the node is in contact with a large heat pipe that supplies heat to the node through an impedance AIPED so as to maintain its temperature near ABS (EFLOW). Large values (approximately 1000) of AIPED are recommended.
AIPED	real	0.	Impedance parameter. If AIPED is nonzero, the code interprets SKD as a flowing wellbore pressure (MPa) with an impedance ABS(AIPED). If AIPED < 0, flow is only allowed out of the well. For heat only, AIPED is the thermal resistance. If AIPED = 0, SKD is flow rate. If AIPED ≠ 0 and SKD = 0, the initial value of pressure will be used for the flowing pressure.
Isothermal air-water			
Case 1: AIPED = 0 (constant mass rate, 1- or 2-phase source or sink)			
SKD	real	0.	Mass source strength (kg/s). Negative value indicates injection into the rock mass.

Input variable	Format	Default	Description
EFLOW	real	0.	a) EFLOW ≥ 0, EFLOW is source liquid saturation. $Q_w = SKD \cdot EFLOW$ (kg/s). $Q_a = SKD \cdot (1 - EFLOW)$ (kg/s). b) EFLOW < 0, ABS(EFLOW) is the source air pressure (MPa). $Q_w = SKD$ (kg/s). $Q_a = 1.0 \cdot (P_a - ABS(EFLOW))$ (kg/s). In the above and following relations, Q_w is the source term for water, Q_a is the source term for air, and P_a is the in-place air pressure. The second case works well in situations in which inflow is specified and it is desired to hold the air pressure at a constant value.
Case 2: AIPED > 0 (constant pressure, constant liquid saturation source or sink)			
SKD	real	0.	Specified source air pressure (MPa).
EFLOW	real	0.	a) EFLOW < 0, air only source. $Q_a = AIPED \cdot (P_a - SKD)$ (kg/s). b) $0 < \mathrm{EFLOW} \le 1$, EFLOW is specified source liquid saturation. For $\mathrm{SKD} \ge 0$, 2-phase source. $Q_a = AIPED \cdot (P_a - SKD)$ (kg/s). $Q_w = AIPED \cdot (S_l - EFLOW)$ (kg/s). For $\mathrm{SKD} < 0$, water only source. $Q_a = 0$. In the above relation, S_l is the in-place liquid saturation.
AIPED	real		Impedance parameter. A large value is recommended (10^2 - 10^6) to create a flow term large enough to maintain constant pressure.
Case 3: AIPED < 0 (Outflow only , if P_l > SKD)			
SKD	real	0.	Pressure above which outflow occurs (MPa).
EFLOW	real	0.	Not used.
AIPED	real	0.	Impedance parameter. $Q_w = ABS(AIPED) \cdot R_l / \mu_l (P_l - SKD)$ (kg/s), where R_l is the water relative permeability and μ_l is the water viscosity.

The following is an example of **flow**

```
flow
   88        88        1        0.050       -25.0          0.
   14        14        1        3.600      -160.0          1.
```

6.2.19 Control statement **flo2** (optional)

Group 1 - JA, JB, JC,JD, SKD, EFLOW, AIPED (SKD, EFLOW, AIPED - defined on page 34 under control statement **flow**)

Multiple lines of input may be used terminated by a blank line.

Input variable	Format	Description
JA	integer	Indices used to define planes in a 3-D simulation with a regular numbering pattern. The flow rates are defined within the inner loop of the do loops: DO JK = JA, JB KL = JK - JA DO IJ = JA + KL, JC + KL, KD . . . ENDDO ENDDO
JB	integer	
JC	integer	
JD	integer	

6.2.20 Control statement **flxo** (optional)

Mass flux between two nodes is output by choosing this control statement.

Group 1 - NFLX

Group 2 - IFLX1, IFLX2 (repeated NFLX times)

Group 3 - X1, Y1, Z1 (as needed)

Group 4- X2, Y2, Z2 (as needed)

Input variable	Format	Description
NFLX	integer	Number of internode fluxes to be calculated.
IFLX1	integer	First node to be used in flux calculation.
IFLX2	integer	Second node to be used in flux calculation.
X1	real	Coordinates of the first node to be used in flux calculation. Used only for those nodes for which IFLX1 < 0.
Y1	real	
Z1	real	
X2	real	Coordinates of the second node to be used in flux calculation. Used only for those nodes for which IFLX2 < 0.
Y2	real	
Z2	real	

If IFLX1 < 0, then after all IFLX1 and IFLX2 values are read, coordinates X1, Y1, and Z1 are read, and the node nearest to these coordinates is used.

If IFLX2 < 0, coordinates for the second node are read in on another line. The code cycles through each IFLX1 and IFLX2 in this manner, reading coordinates when needed. Results are written to the screen, if tty output is enabled, and to the output file **iout**.

6.2.21 Control statement hflx (optional)

Group 1 - JA, JB, JC, QFLUX, QFLXM (JA, JB, JC - defined on page 19)

A negative heat flux indicates heat flow into the reservoir.

Input variable	Format	Default	Description
QFLUX	real	0.	If QFLXM = 0, then QFLUX is the heat flux (MW). If QFLXM ≠ 0, then QFLUX is a temperature (°C) and the heat flux is calculated according to the formula: $Q_H = QFLXM(TL - QFLUX)$ (MW).
QFLXM	real	0.	If QFLXM ≠ 0, multiplier for heat flux equation given in QFLUX description (MW/°C). This must be large for large temperature gradients or when a constant temperature must be maintained.

The following is an example of **hflx**

```
hflx
  401         410           1           -0.001      0.0
```

6.2.22 Control statement ice (optional)

Ice-phase calculations, not tested.

Group 1 - ICE, SIIN, TMELT

Group 2 - JA, JB, JC, SII (JA, JB, JC - defined on page 19)

Input variable	Format	Description
ICE	integer	Solution descriptor for ice solution. ICE = 0, information is read but not used. ICE ≠ 0, **ice** solution is implemented.
SIIN	real	Default value for ice saturation (used when ice saturation SII in Group 2 is set to 0 at any node).
TMELT	real	Freezing temperature of water (°C).
SII	real	Ice saturation. The default value is [0].

6.2.23 Control statement init (required if macro pres not used)

Set initial pressure and temperature at all nodes.

Group 1 - PEIN, TIN, TIN1, GRAD1, DEPTH, TIN2, GRAD2, QUAD

Note that the macro **pres** may overwrite some of the values that are set by macro **init**.

Input variable	Format	Description
PEIN	real	Initial value of pressure (MPa). If initial values are read from the read file (iread), then this value is ignored. If gravity is present, this is the value of the pressure at node 1, and the other nodal pressures are adjusted by applying the hydraulic head. Absolute pressures are used. Pressure as a function of depth is calculated with TIN < 0.
TIN	real	Initial value of temperature (°C). If TIN ≤ 0, then the initial temperatures are calculated using the temperature-gradient formulas given below.
TIN1	real	Defined in formulas below (°C).
GRAD1	real	Defined in formulas below (°C/m).
DEPTH	real	Defined in formulas below (m).
TIN2	real	Defined in formulas below (°C).
GRAD2	real	Defined in formulas below (°C/m).
QUAD	real	Defined in formulas below (°C/m^2).

$$T = TIN1 + GRAD1 \times Z \qquad 0 \le \mathrm{Z} \le \mathrm{DEPTH}$$

$$T = TIN1 + GRAD2 \times Z + QUAD \times Z^2 \qquad \mathrm{Z} > \mathrm{DEPTH}$$

The following is an example of **init**

```
init
    3.6         0.0        240.        0.         0.         240.        0.         0.
```

6.2.24 Control statement iter (optional)

If the user is not familiar with the linear-equation-solver routines in FEHM, control statement **iter** should not be used.

Group 1 - G1, G2, G3, TMCH, OVERF

Group 2 - IRDOF, ISLORD, IBACK, ICOUPL, RNMAX

Input variable	Format	Default	Description
G1	real	1.e-6	Multiplier for the linear-convergence region of the Newton-Raphson iteration.
G2	real	1.e-6	Multiplier for the quadratic-convergence region of the Newton-Raphson iteration.
G3	real	1.e-3	Tolerance for the adaptive implicit method (multiplying factor for Newton-Raphson tolerance).
TMCH	real	1.e-9	Machine tolerance. If satisfied by the residual norm, the Newton iteration is assumed to be complete. If TMCH is < 0, the ABS(TMCH) is used as a tolerance for each equation at each node. Convergence is achieved if the residual of every equation at every node is < ABS(TMCH).

Input variable	Format	Default	Description
OVERF	real	1.1	Over-relaxation factor for passive nodes in adaptive implicit method.
IRDOF	integer	0	Enables the reduced degree-of-freedom method. [0] Set to 0 if reduced degrees of freedom are not required. When IRDOF = 1, a reduced degree of freedom from 3 to 2 or 3 to 1 is used. When IRDOF = 2, a reduced degree of freedom from 3 to 2 is used. If IRDOF=11, then an air-only solution is found for the isothermal air-water process model.
ISLORD	integer	0	Reordering parameter. The ordering can be understood by labeling the mass equation as 1, the heat equation as 2, and the noncondensible gas equation (if it exists) as 3. The value of ISLORD and the corresponding equation order is given below. The ordering has an effect on the speed of convergence of several solution algorithms but will not affect most users.

ISLORD	2 Degrees of Freedom	3 Degrees of Freedom	4 Degrees of Freedom
0	1, 2	1, 2, 3	1, 2, 3, 4
1	2, 1	1, 3, 2	1, 3, 2, 4
2		2, 1, 3	
3		2, 3, 1	
4		3, 1, 2	
5		3, 2, 1	

Input variable	Format	Default	Description
IBACK	integer		IRDOF parameter. [0] If IBACK = 1, SOR iterations are performed before call to solver. If IBACK = 2, SOR iterations are performed before call to SOLVER, and SOLVER is called twice.
ICOUPL	integer		Number of SOR iterations used in reduced degree of freedom methods. [0]
RNMAX	real		Maximum running time for problem before the solution is stopped (cpu minutes) (very large if not set with control statement **iter**).

The following is an example of **iter**

```
iter
    1.e-5          1.e-5          1.e-5          1.e-9          1.2
    1              0              0              2              200.0
```

6.2.25 Control statement itup (optional)

Group 1 - IAD_UP

Input variable	Format	Default	Description
IAD_UP	integer	100	Number of iterations after which the upwind directions are held constant. [2]

6.2.26 Control statement **iupk** (optional)

No input is associated with this control statement. If enabled, the full transmissibility term will be upwinded (including the intrinsic permeability). Otherwise, the fluid and relative-permeability part of the transmissibility will be upwinded and the intrinsic permeability will be harmonically averaged.

6.2.27 Control statement **ivfc** (optional)

Enables volume-control subroutine. Not supported in this version.

6.2.28 Control statement **mdnode** (optional)

Enables extra connections to be made to nodes, which is useful for simulating wellbore connections, faults, and flow across internal boundaries.

Group 1 - NUM_MD, MAX_CON

Group 2 - NODE, IPAR, NPAR (repeated NUM_MD times)

Input variable	Format	Default	Description
NUM_MD	integer	0	Number of new connections to be entered.
MAX_CON	integer	0	Maximum number of new connections to a given node. This number does not include old connections. Thus, if a node was already connected to 5 neighboring nodes and two new connections were added to this node in this macro statement and this was the maximum number of connections added in this macro statement, then MAX_CON = 2.
NODE	integer	0	Node to which new connection is established.
IPAR	integer	0	IPAR is not used at present. Its value is ignored. However the entered number must be an integer.
NPAR	integer	0	NPAR is the new connected node. If NPAR = NODE, no new connection is established.

The following are examples of **mdnode**

```
mdnode
   3        2
  10        0         15
 100        0        106
  10        0        320
```

```
mdnode
   4        3
   1        0         16
   2        0          1
   4        0          1
  10        0        203
```

6.2.29 Control statement **ngas** (optional)

Noncondensible gas transport.

Group 1 - ICO2D

Group 2 - JA, JB, JC, PCO2 (JA, JB, JC - defined on page 19)

Group 3 - JA, JB, JC, CPNK (JA, JB, JC - defined on page 19)

Group 4 - JA, JB, JC, QCD (JA, JB, JC - defined on page 19)

Input variable	Format	Default	Description
ICO2D	integer	3	Solution descriptor for noncondensible gas transport. ICO2D = 1, the 3-degree-of-freedom solution will be reduced to a 1-degree-of-freedom problem. (See macro **iter**, the parameter ICOUPL is also set to 5 if ICO2D = 1.) ICO2D = 2, the 3-degree-of-freedom solution will be reduced to a 2-degree-of-freedom problem. (See macro **iter**, the parameter ICOUPL is also set to 5 if ICO2D = 2.) ICO2D = 3, full 3 degrees of freedom.
PCO2	real	0.	Initial partial pressure of noncondensible gas. If PCO2 < 0, then ABS (PCO2) is interpreted as a temperature and the partial pressure of the noncondensible gas is calculated according to the formula: $PCO2 = P_T - P_{SAT}(T)$, where P_T is the total pressure and $P_{SAT}(T)$ is the water-saturation pressure and is a function of temperature only.
CPNK	real	0.	If CPNK < 0, then ABS (CPNK) is the specified noncondensible pressure and will be held at that value. If CPNK > 0, then CPNK is the specified relative humidity and the saturation, S_l, is calculated using the vapor-pressure lowering formula and the capillary-pressure formula: $Pcap(S_l) = \ln(h)\rho_l RT$, where $Pcap$ is the capillary function, h is the humidity, R is the gas constant, T is the temperature, and ρ_l is the liquid density. Once the formula is solved, S_l is held constant.
QCD	real	0.	Specified air flow rate (kg/sec).

The following is an example of **ngas**

```
ngas
   3
   1        800        1      -20

   1        800        1        0.

   1        800        1        0.

```

6.2.30 Control statement nod2 (optional)

Specify the node numbers for which detailed file output is desired and alternate nodes for terminal output.

Group 1 - M, M2

Group 2 - MN (1), MN (2), . . . , MN (M)

Group 3 - X, Y, Z (as needed)

Group 4 - MNI(1), MNI(2), . . . , MNI(M2)

Group 5 - X, Y, Z (as needed)

Input variable	Format	Description
M	integer	Number of nodes for which information will be printed on the output file (iout). If M ≤ 0, pressure and temperature will be written on the output file for all nodes, but no nodal parmeter values will be printed in the history plot files. Group 2 is omitted if M ≤ 0.
M2	integer	Number of nodes for short list (terminal printout). If M2 ≤ 0, Group 4 is omitted.
MN	integer	M node numbers for which information will be printed on the output file (iout). If a MN(I) < 0, then coordinates are used to define that print-out node, and the coordinate sets (X, Y, Z) for each MN(I) < 0 are added after Group 2.
MNI	integer	M2 node numbers for which information will be printed on the terminal (short list). This group exists only if M2 ≠ 0. If MNI(I) < 0, then coordinates are used to define the terminal output nodes, and the coordinate sets (X, Y, Z) for each MNI(I) < 0 are added after Group 4.
X	real	Coordinates of node for which information will be printed. One line for each MN or MNI < 0. The code finds the node closest to the coordinate given. For 2-D problems, set Z = 0. No input if no MN or MNI < 0.
Y	real	
Z	real	

The following are examples of **nod2**

```
nod2
    2         1
   50        88
   50
```

```
nod2
    2         1
   50        88
  -88
  100.      1000.        0.
```

6.2.31 Control statement node (optional)

Specify the node numbers for which detailed output is desired.

Group 1 - M

Group 2 - MN (1), MN (2) ... MN (M)

Group 3 - X, Y, Z (as needed)

or

Group 1 - KEYWORD

Group 2 - JA, JB, JC (JA, JB, JC - defined on page 19)

Input variable	Format	Description
M	integer	Number of nodes for which information will be printed on the output (iout) and history plot (ishis, istrc) files. If M ≤ 0, pressure and temperature will be written on the output file for all nodes, but no nodal parmeter values will be printed in the history plot files. Group 2 is omitted if M ≤ 0.
MN	integer	M node numbers for which information will be printed on the output file (iout). If MN(I) < 0, then coordinates are used to define the print-out node, and the coordinate sets (X, Y, Z) for each MN(I) < 0 are added after Group 2.
X	real	Coordinates of node for which information will be printed. One line for each MN < 0. The code finds the node closest to the coordinate given. For 2-D problems, set Z = 0. No input if no MN >0.
Y	real	
Z	real	
KEYWORD	character*5	Key word for invoking node specification by ja, jb, jc format. The necessary word is *block*.

The following are examples of **node**

```
node
    2
  50          88
```

```
node
    2
  50          -88
 100.        1000.          0.
```

```
node
block
   1          100          10
  -3            0           0
```

6.2.32 Control statement num (Not implemented)

6.2.33 Control statement perm (required)

Assign permeabilities of the rock. Permeabilities represent average values of a volume associated with a node. Note that using **rlp** to describe relative permeabilities causes these values to be overwritten.

Group 1 - JA, JB, JC, PNXD, PNYD, PNZD (JA, JB, JC - defined on page 19)

Input variable	Format	Default	Description
PNXD	real	1.e-30	Permeability in the x-direction (m^2).
PNYD	real	1.e-30	Permeability in the y-direction (m^2).
PNZD	real	1.e-30	Permeability in the z-direction (m^2).

The following is an example of the **perm** macro

```
perm
   1        140          1        2.50e-14        2.50e-14        0.00e-00
```

6.2.34 Control statement ppor (optional)

Group 1 - IPOROS, R1, R2, R3

Group 2 - JA, JB, JC, R4, R5 (JA, JB, JC - defined on page 19)

Input variable	Format	Default	Description
IPOROS	integer		Porosity/permeability type. IPOROS = 0, constant porosity. IPOROS = 1, simple linear model: $\phi = \phi_0 + (1 - \phi_0)(c_r - c_g)(P - P_0)$. IPOROS = 2, Gangi stress model: $\phi = \phi_0\left[1 - \left(\frac{P_c}{P_0}\right)^m\right]$ and $P_c = \sigma - P - \alpha E \Delta T$. This option should only be used with single-phase liquid-water problems.
R1	real		Parameter used in the linear and Gangi models. For the linear model, pore volume compressibility, c_r (MPa $^{-1}$). For the Gangi model, coefficient of thermal expansion, α (°C $^{-1}$).
R2	real		Parameter used in the linear and Gangi models. For the linear model, compressibility of the matrix grain, c_g (MPa $^{-1}$). For the Gangi model, Young's modulus, E (MPa).
R3	real		Parameter used in the Gangi model, initial stress, σ (MPa).
R4	real	1	Variable parameter used in Gangi model, exponent m.
R5	real	1.e30	Variable parameter used in Gangi model, pressure P_0 (MPa).

For the linear model, P_0 is PEIN from macro **init**, ϕ_0 is PSD from macro **rock,** and k_0 is from macro **perm**. For either porosity model, permeability is given by $k = k_0\left(\frac{\phi}{\phi_0}\right)^3$.

6.2.35 Control statement pres (required if macro init not used)

Group 1 - JA, JB, JC, PHRD, TIND, IEOSD (JA, JB, JC - defined on page 19)

The initial values defined in control statement **pres** supersede all others. Note that the term "saturated" is a thermodynamic definition and not the groundwater hydrology definition (volumetric fraction of pore void that is filled with water: IEOSD = 1). Saturated here indicates that vapor and liquid phases exist simultaneously. The superheated region means that all pore space is filled with gas.

Input variable	Format	Default	Description
PHRD	real	PEIN	Initial pressure (MPa).
TIND	real		Initial temperature (°C) if IEOSD = 1 or 3, initial saturation if IEOSD = 2

IEOSD	integer	1	Thermodynamic-region parameter. If IEOSD < 0, then the code uses ABS (IEOSD) and fixes the values of PHRD and TIND to the values provided above.
			IEOSD = 1, the compressed-liquid region.
			IEOSD = 2, the saturation region.
			IEOSD = 3, the superheated region.

The following is an example of **pres**

```
pres
   -1        0        1        0.1      0.1        2
   -2        0        1        0.1      0.1        2
   -3        0        1        0.1      0.003      2
   -4        0        1        0.1      0.1        2
   -5        0        1        0.1      0.11       2
   -6        0        1        0.1      0.11      -2
    1      800        1        0.1      0.5        2
```

6.2.36 Control statement ptrk (optional, cannot be used with trac)

Group 1 - NPART, RSEED

Group 2 - DAYCS, DAYCF, DAYHF, DAYHS

Group 3 - TRAK_TYPE, HALF_LIFE, POUT, PRNT_RST

Group 4 - TRANSFLAG(JJ), KD(JJ), TCLX(JJ), TCLY(JJ), TCLZ(JJ), DIFFMAT(JJ), RD_FRAC(JJ), MATRIX_POR(JJ), FSPACING(JJ)

Group 5 - JA, JB, JC, ITRC (JA, JB, JC - defined on page 19)

Group 6- JA, JB, JC, PCNSK, T1SK, T2SK (JA, JB, JC - defined on page 19)

Group 4 is used to define models in which identical sorption and transport parameters are assumed to apply. Group-4 data are read until a blank line is encountered. The model number JJ is incremented by 1 each time a line is read.

The concentration output is written to the .trc, .out, AVS concentration output files, and the .fin file, if specified (nonzero value of PRNT_RST).

Input variable	Format	Description
NPART	integer	Number of particles in the simulation. Note: the actual number may be slightly less than the number specified by the user because when the code divides the particles among the starting nodes as specified in Group 7, the code must input an integer number of particles at each node.
RSEED	integer	6-digit integer random number seed.

Input variable	Format	Description
DAYCS	real	Time that the particle-tracking solution is enabled (days).
DAYCF	real	Time that the particle-tracking solution is disabled (days).
DAYHF	real	Time that the flow solution is disabled (days).
DAYHS	real	Time that the flow solution is enabled (days).
TRAK_TYPE	integer	Flag to denote the fluid phase of the particles. 1 - liquid-phase particles. 2 - vapor-phase particles.
HALF_LIFE	real	Half-life for irreversible first-order decay reaction(s). Set HALF_LIFE = 0 for no decay.
POUT	integer	Flag to specify the concentration output. 1 - Concentrations computed as number of particles per unit total volume (rock and fluid). 2 - Concentrations computed as number of particles per unit fluid volume (the fluid is liquid for TRAK_TYPE = 1 and gas for TRAK_TYPE = 2). 3 - Concentrations computed as number of particles at a given node point. 4 - Used for radioactive-particle mixing model (only liquid tracer). For meaningful results, the particles must all be injected simultaneously in a pulse (give a very short duration of injection starting at time 0). The file getconc.f contains data describing the function f(t) vs. time, where f(t) is given as $\int_0^t C(t) \exp(-kt) dt$ The meaningful output are the final concentrations after all the particles have left the system. -1, -2, -3, or -4 - Concentrations computed as specified above for abs(pout). The .trc file contains breakthrough output for the first node specified in the node macro. 0 - Concentration output is a running total of the number of particles that have left each node divided by the fluid or vapor mass at that node, depending on trak_type.

Input variable	Format	Description
PRNT_RST	integer	Flag to specify whether particle information is written to the ".fin" file. 0 - Particle information is not written to ".fin" file. 1 - Particle information is written to the ".fin" file. -1 - Particle positions and ages are written to the ".fin" file. When particle-tracking data are written to the .fin file, the arrays are written after all of the heat- and mass-simulation information. The information written is sufficient to perform a restart of the particle-tracking simulation and to postprocess the data to compile statistics on the particle-tracking run. However, for a large number of particles, this file can become quite large, so particle-tracking information should only be written when necessary. Thus, 0 should be used for PRNT_RST unless restarting or postprocessing to obtain particle statistics is required. Selecting the -1 option allows a subset of the full set of information needed for a restart (particle positions and ages) to be written. Restart runs that use this file as input will only be approximate, because the particle is assumed to have just entered its current cell. For restart runs, PRNT_RST = 1 is preferred, whereas PRNT_RST = -1 is appropriate for output of particle statistics for postprocessing.
TRANSFLAG	integer	Flag to specify which transport mechanisms apply. 1 - advection only (no dispersion or matrix diffusion). 2 - advection and dispersion (no matrix diffusion). 3 - advection and matrix diffusion (no dispersion). 4 - advection, dispersion, and matrix diffusion.
KD	real	Sorption coefficient (linear, reversible, equilibrium sorption). Units are kg-fluid/kg-rock (these units are equivalent to the conventional units of cc/g when the carrier fluid is water at standard conditions). This value applies to the medium as a whole when matrix diffusion is turned off, whereas for simulations invoking matrix diffusion, the value applies to the rock matrix. For the latter case, sorption in the flowing system (fractures) is modeled using the RD_FRAC variable.
TCLX	real	Dispersivity in the x-direction (m). The input value is ignored when dispersion is turned off.
TCLY	real	Dispersivity in the y-direction (m). The input value is ignored when dispersion is turned off.
TCLZ	real	Dispersivity in the z-direction (m). The input value is ignored when dispersion is turned off.
DIFFMAT	real	Molecular diffusion coefficient in the rock matrix (m^2/s). The input value is ignored unless matrix diffusion is invoked.
RD_FRAC	real	Retardation factor within the primary porosity (fractures) for a matrix-diffusion particle-tracking simulation (use 1 for no sorption on fracture faces). The input value is ignored unless matrix diffusion is invoked.

Input variable	Format	Description
MATRIX_POR	real	Porosity of the rock matrix. Used to simulate diffusion and sorption in the rock matrix when matrix diffusion is invoked. Note: when matrix diffusion is turned off, particle transport through the medium is computed using the porosity set in the **rock** macro, and the input value of MATRIX_POR is ignored.
FSPACING	real	Mean fracture spacing (m). When matrix diffusion is invoked, the mean fracture aperture (a parameter in the matrix-diffusion model) is computed as fracture porosity (from the **rock** macro) divided by FSPACING. When matrix diffusion is turned off, the value of FSPACING is ignored.
ITRC	integer	Model number for parameters defined in group 4. Default is [1].
PCNSK	real	Particle-injection parameter assigned for nodes defined by JA, JB, and JC. Two options are available. PCNSK > 0 - particles are injected at each node in proportion to the source mass flow rate at the node. When multiple lines of input are given for Group 6, PCNSK is proportional to the particle-injection concentration. This boundary condition is equivalent to injecting a solute of a given concentration into the system. Note: the source flow rates used to assign the number and timing of particle injections are those at the beginning of the particle-tracking simulation (time DAYCS). Transient changes in this source flow rate during the particle-tracking simulation do not change the input of particles to the system. PCNSK < 0 - particles are introduced at the node(s), regardless of whether there is a fluid source at the node. When multiple lines of input are given for Group 6, abs(PCNSK) is proportional to the number of particles introduced at the node(s). When multiple lines of input are given for Group 6, all PCNSK values must have the same sign (i.e., the two options cannot be invoked in the same simulation). Default is 0 for all unassigned nodes, meaning that no particles are injected at that node.
T1SK	real	Time (days) when particle injection begins. Default is [0].
T2SK	real	Time (days) when particle injection ends. Default is [0].

Notes on Restarting: As with all restart runs for FEHM, a .ini file is specified to be read to set the initial conditions upon restarting. However, there are two possibilities for restart calculations with particle tracking: 1) the heat- and mass-transfer solution is being restarted, but the particle-tracking simulation is initiated during the restart run (it was not carried out in the simulation that generated the .ini file); or 2) the heat- and mass-transfer solution and the particle-tracking simulation are both being restarted. If the code does not find the "ptrk" key word at the top of the .ini file, then the original run did not employ particle tracking, and Case 1 is assumed. A common example is a preliminary calculation that establishes a fluid-flow steady state, followed by a restart simulation of transport.

If "ptrk" was written into the .ini file in the original run, the particle data in the .ini file are read and used to initialize the particle-tracking simulation (Case 2). In this instance, the number of particles (NPART) must be set the same for the restart run as in the original run, or the results will be unpredictable. When restarting a particle-tracking simulation, certain input data are overwritten by information in the .ini file. These parameters include RSEED, PCNSK, T1SK, and T2SK. Other input parameters can be set to different values in the restart run than they were in the original run, but of course, care must be taken to avoid physically unrealistic assumptions, such as an abrupt change in transport properties of Group 4 part way through a simulation.

A final note on restart calculations is in order. A common technique in FEHM restart calculations is to reset the time at the top of the .ini file to 0, so that the starting time of the restart simulation is arbitrarily 0 rather than the ending time of the original simulation. This technique is useful for the example of the steady-state-flow calculation, followed by a restart solute-transport calculation. Although the technique is acceptable for particle-tracking runs that are initiated only upon restart (Case 1), it is invalid when a particle-tracking run is being resumed (Case 2). The reason is that all particle times read from the .ini file are based on the starting time of the original simulation during which the particle-tracking simulation was initiated.

The following is an example of **ptrk**:

```
ptrk
  100000     122945
    10.        20.        10.        20.
     1          0          2          0
     4          0.         2.         2.         2.        5.e-11      1.        0.1       0.333
     4          3.         2.         2.         2.        1.e-10      1.        0.28      2.

     1          0          0          1
    -2          0          0          2

    -3          0          0          1.        10.        10.0001

```

In this example, 100,000 nondecaying, liquid-borne particles are introduced as a sharp pulse (from time 10 to 10.0001 days) with the injection fluid in zone 3 (an injection well defined in the **zone** macro preceeding **ptrk**). The particle-tracking simulation starts as the heat- and mass-transfer simulation is turned off at day 10, after having established a fluid-flow steady state. Two models are defined for assigning transport properties of the particles. All nodes are assigned to model 1, after which model-2 properties are assigned for zone 2. A combined advection, dispersion, and matrix-diffusion model is used for all nodes. However, sorption in the matrix occurs only for model 2 (which is zone 2 in this simulation), and the matrix-transport properties (porosity, fracture spacing, diffusion coefficient) differ for this model as well.

6.2.37 Control statement renm (optional)

Group 1 - JA, JB, JC, IGD (JA,JB,JC - defined on page 19)

Input variable	Format	Description
IGD	integer	New node number for given node.

6.2.38 Control statement rflx (optional)

Radiation-heat source term. Not implemented in this version. A negative heat flux indicates heat flow into the reservoir.

Group 1 - EMISS

Group 2- JA, JB, JC, QFLUX, QFLXM (JA, JB, JC - defined on page 19)

Input Variable	Format	Description
EMISS	real	Emissivity.
QFLUX	real	If QFLXM = 0, then QFLUX is the heat flux (MW). If QFLXM ≠ 0, then QFLUX is a temperature, and the heat flux is calculated according to the formula: $Q_H = QFLXM(TL - QFLUX)$ (MW). [0]
QFLXM	real	Multiplier for heat flux equation given in QFLUX description (MW/°C). If QFLXM = 0, then QFLUX is the heat flux (MW). [0]

6.2.39 Control statement rlp (optional)

Relative-permeability and capillary-pressure model. Four models are available. The fifth model flag is used to designate that **rlp** data should be read from an auxiliary file.

Group 1 - IRLP(i), RP1, RP2, RP3, RP4, RP5, RP6, RP7, RP8, RP9, RP10, RP11, RP12, RP13, RP14, RP15 (number of parameters depends on model selected)

Group 2 - JA, JB, JC, I (JA, JB, JC - defined on page 19)

Only those parameters defined for a given model need to be input. Group 1 is ended when a blank line is encountered. The parameter *i* is incremented each time a Group 1 line is read. Group 2 lines will refer to this parameter. For model number 4 (the combined van Genuchten model), the permeability is isotropic and overwrites the input from macro **perm**.

or

Group 1 - IRLP(i)

Group 2 - RLPFILE

Input variable	Format	Description
IRLP(i)	integer	Relative-permeability model type.
Model 1: IRLP(i) = 1, linear relative permeability, linear capillary pressure (6 parameters required).		
RP1	real	Residual liquid saturation.
RP2	real	Residual vapor saturation.
RP3	real	Maximum liquid saturation.
RP4	real	Maximum vapor saturation.
RP5	real	Capillary pressure at zero saturation (Mpa).
RP6	real	Saturation at which capillary pressure goes to zero.
Model 2: IRLP(i) = 2, Corey relative permeability, linear capillary pressure (4 parameters required).		
RP1	real	Residual liquid saturation.
RP2	real	Residual vapor saturation.
RP3	real	Capillary pressure at zero saturation (MPa).
RP4	real	Saturation at which capillary pressure goes to zero.
Model 3: IRLP(i) = 3, van Genuchten relative permeability, van Genuchten capillary pressure (6 parameters required).		
RP1	real	Residual liquid saturation.
RP2	real	Maximum liquid saturation.
RP3	real	Inverse of air entry pressure, α_G (1/m). (Note: some data are given in (1/Pa); convert using pressure = $\rho g \Delta h.$)
RP4	real	Power n in van Genuchten formula.
RP5	real	Low-saturation fitting parameter, multiple of cutoff capillary pressure assigned as maximum capillary pressure. If RP5 < 0, then a linear fit from this cutoff saturation (RP6 is used). The slope of the cutoff saturation is used to extend the function to saturation = 0. If RP5 = 0, a cubic fit is used. The slope at the cutoff saturation is matched, and the conditions $\frac{\partial}{\partial S}Pcap = 0$ and $\frac{\partial^2}{\partial S}Pcap = 0$ are forced at $S = 0$. If RP5 > 0, a multiple of the value of the capillary pressure at the cutoff saturation, $RP5 \bullet Pcap(S_{cutoff})$, is forced at $S = 0$.
RP6	real	Cutoff saturation used in fits described for RP5. Must be greater than RP1.

Input variable	Format	Description
Model 4: IRLP(i) = 4, van Genuchten relative permeability, van Genuchten capillary pressure, effective continuum (15 parameters required).		
RP1	real	Residual liquid saturation, matrix rock material.
RP2	real	Maximum liquid saturation, matrix rock material.
RP3	real	Inverse of air entry pressure, α_G (1/m), matrix rock material. (Note: some data are given in (1/Pa); convert using pressure = $\rho g \Delta h$.)
RP4	real	Power n in van Genuchten formula, matrix rock material.
RP5	real	Low-saturation fitting parameter, matrix rock material, multiple of cutoff capillary pressure assigned as maximum capillary pressure. If RP5 < 0, then a linear fit from this cutoff saturation (RP6 is used). The slope of the cutoff saturation is used to extend the function to saturation = 0. If RP5 = 0, a cubic fit is used. The slope at the cutoff saturation is matched, and the conditions $\frac{\partial}{\partial S}Pcap = 0$ and $\frac{\partial^2}{\partial S}Pcap = 0$ are forced at $S = 0$. If RP5 > 0, a multiple of the value of the capillary pressure at the cutoff saturation, $RP5 \bullet Pcap(S_{cutoff})$ is forced at $S = 0$.
RP6	real	Cutoff saturation used in fits described for RP5, matrix rock material. Must be greater than RP1.
RP7	real	Residual liquid saturation, fracture material.
RP8	real	Maximum liquid saturation, fracture material.
RP9	real	Inverse of air entry pressure, α_G (1/m), fracture material. (Note: some data are given in (1/Pa); convert using pressure = $\rho g \Delta h$.)
RP10	real	Power n in van Genuchten formula, fracture material.
RP11	real	Low-saturation fitting parameter, fracture material, multiple of cutoff capillary pressure assigned as maximum capillary pressure. If RP11 < 0, then a linear fit from this cutoff saturation (RP6 is used). The slope of the cutoff saturation is used to extend the function to saturation = 0. If RP11 = 0, a cubic fit is used. The slope at the cutoff saturation is matched, and the conditions $\frac{\partial}{\partial S}Pcap = 0$ and $\frac{\partial^2}{\partial S}Pcap = 0$ are forced at $S = 0$. If RP11 > 0, a multiple of the value of the capillary pressure at the cutoff saturation, $RP11 \bullet Pcap(S_{cutoff})$, is forced at $S = 0$.
RP12	real	Cutoff saturation used in fits described for RP11, fracture material. Must be greater than RP7.

Input variable	Format	Description
RP13	real	Fracture permeability (m^2). This is the permeability of the individual fractures. The bulk permeability of the fracture continua is $RP13 \times RP15$.
RP14	real	Matrix-rock saturated permeability (m^2).
RP15	real	Fracture porosity.
Model 5: IRLP(i) = 5, a file containing the relative-permeability and capillary models is required. The name of this file is given on the line following IRLP(i) = 5. A model must be entered for each and every node in numerical (nodal) order. The models above are available. (See example.)		
I	integer	Number referring to the sequence of models read in Group 1. The default is [1].
RLPFILE	character*100	Name of the optional **rlp** data file.

The following are examples of **rlp**

```
rlp
   2          0.3        0.1        2.0        1.

   1        140          1          1

```

```
rlp
   5
  rlp.dat
```

File rlp.dat:

```
   2          0.3        0.1        2.         1.
   2          0.3        0.1        2.         1.
   .           .          .          .          .
   .           .          .          .          .
   .           .          .          .          .
   2          0.3        0.1        2.         1.
```

A model is entered for each node

6.2.40 Control statement rock (required)

Assign rock density, specific heat, and porosity.

Group 1 - JA, JB, JC, DENRD, CPRD, PSD (JA,JB,JC - defined on page 19)

Input variable	Format	Description
DENRD	real	Rock density (kg/m^3).
CPRD	real	Rock specific heat ($\frac{MJ}{kg \cdot K}$). If CPRD > 1, the code will assume the units are ($\frac{J}{kg \cdot K}$) and multiply by 10^{-6}.
PSD	real	Porosity.

The following is an example of **rock**

```
rock
    1          140            1         2563.           1010.             0.3500
```

6.2.41 Control statement rxn (optional)

Chemical reactions between species are invoked with this control statement. It is used in conjunction with control statement **trac**.

Group 1 - KEY_GROUP

Group 2 - NGROUPS

Group 3 - GROUP(ISPECIES), ISPECIES = 1, NSPECI (repeated NGROUPS times, once for each group)

Group 4 - NRXNS, RXN_INTERVAL

For each reaction, there are two possibilities that require different input: equilibrium reaction (KEY_RXN = “equilibrium”) and kinetic reaction (KEY_RXN = “kinetic”). These two choices may be mixed in a given simulation to simulate a combination of equilibrium and kinetic reactions. Group-5 parameters are repeated NRXN times, once for each reaction. Within an equilibrium reaction option, there are also two options based on the choice of the EQUIL_MODEL parameter. For EQUIL_MODEL = 1, the following input is used:

Group 5 - KEY_RXN, EQUIL_MODEL, EQUIL_CONST25, ENTHALPY25, GAMMA_CHECK, RATE_FACTOR, ROUND_TOL

For EQUIL_MODEL = 2, the input is:

Group 5 - KEY_RXN, EQUIL_MODEL, AWWA(1), AWWA(2), AWWA(3), AWWA(4), AWWA(5), GAMMA_CHECK, RATE_FACTOR, ROUND_TOL

Finally, for a kinetic reaction, the input is:

Group 5- KEY_RXN, AR_FOR, EA_FOR, AR_REV, EA_REV

Group 6 - STOIC(ISPECIES), ISPECIES = 1, NSPECI (repeated NRXN times, once for each reaction)

Group 7 - RATE_POWER(ISPECIES), ISPECIES= 1, NSPECI (repeated NRXN times, once for each reaction)

Group 8 - FL_MULT(ISPECIES), ISPECIES= 1, NSPECI (repeated NRXN times, once for each reaction)

Group 9 - SB_MULT(ISPECIES), ISPECIES= 1, NSPECI (repeated NRXN times, once for each reaction)

Group 10 -H_MULT(ISPECIES), ISPECIES= 1, NSPECI (repeated NRXN times, once for each reaction)

Input for Groups 8 and 9 may be omitted by entering a blank line after the entries for Group 7. In this case, all reactions are assumed to occur with both fluid- and sorbed-phase solutes. If this is the case, or if there are no sorbing solutes in the simulation, then Groups 8 and 9 need not be entered. To specify the nature of the reactions involving sorbed-phase solutes (i. e., whether to compute the rate based on fluid concentration, sorbed-phase concentration, or both), see the variable descriptions below. Note that if any values in Group 8 or 9 are to be set other than to their default values, then all values for these two groups are required.

If there are no Henry's Law species present in the simulation, then Group 10 should be omitted. However, this group is necessary when a Henry's Law species is present, even if it does not participate in any reactions.

Input variable	**Format**	**Description**
KEY_GROUP	character	Key word to specify that species are to be placed into groups that are solved simultaneously. If "gr" is found as the first two characters of the line immediately after the **rxn** control statement, then the code reads in the species group information (Groups 2 and 3) of the input. Otherwise, the code assumes that each species is to be solved separately with an outer iterative loop to ensure overall convergence.
NGROUPS	integer	Number of groups of species. The species assigned to a group (using the GROUP input below) are solved simultaneously. If there is only one group, the solution is complete when convergence is achieved. If there is more than one group, an outer iterative loop over all groups is employed to ensure overall convergence.
GROUP	integer	NSPECIES values are entered for each line of input, and NGROUPS lines of input are required, one for each group. If a value is nonzero, then that species is present in the group. A value of zero denotes that the species is not present in the group. No more than four species can be assigned to any group, but there is no restriction on the number of groups that a species can be assigned to. Grouping of species that take part in rapid kinetics or equilibrium reactions is required for convergence. However, memory requirements increase as the square of the maximum number of species in a group.
NRXNS	integer	Number of chemical reactions.

Input variable	Format	Description
RXN_INTERVAL	real	This parameter allows the user to choose between two different iteration schemes as follows [see "Models and Methods Summary" (Zyvoloski et al. 1997) for a description of these solution procedures]. RXN_INTERVAL = 0 - Solution scheme 1 always used. RXN_INTERVAL > 0 - Solution scheme 2 will be used RXN_INTERVAL times between each use of scheme 1.
KEY_RXN	character	Denotes the type of reaction. The first letter of the keyword is all that is required, but it must appear as the first character of the line. "equilibrium" - equilibrium reaction. "kinetic" - kinetic reaction.
EQUIL_MODEL	integer	Flag denoting which model is to be used for defining the temperature dependence of the equilibrium constant. 1 - van't Hoff model. 2 - Multiparameter fit to experimental data for carbonate system.
EQUIL_CONST25	real	The term A_{rxn} in equilibrium constant-temperature-dependence model.
ENTHALPY25	real	The term ΔH_H in equilibrium constant-temperature-dependence model.
GAMMA_CHECK	real	Equilibrium tolerance parameter γ_{tol}. The reaction will be required to be at equilibrium to within γ_{tol} at every node. For example, a value of 10^{-2} (recommended value) means that the reaction is at least 99% to equilibrium everywhere.
RATE_FACTOR	real	Parameter for scaling the rate constants used in the code for simulating equilibrium behavior. Recommended value: 10^{-3}.
ROUND_TOL	real	Cut-off parameter for forward reaction rate below which the check for equilibrium behavior is not made. Recommended value: 10^{-10}.
AWWA(1)	real	The term $A_{rxn,1}$ in equilibrium constant-temperature-dependence model.
AWWA(2)	real	The term $A_{rxn,2}$ in equilibrium constant-temperature-dependence model.
AWWA(3)	real	The term $A_{rxn,3}$ in equilibrium constant-temperature-dependence model.
AWWA(4)	real	The term $A_{rxn,4}$ in equilibrium constant-temperature-dependence model.
AWWA(5)	real	The term $A_{rxn,5}$ in equilibrium constant-temperature-dependence model.

Input variable	Format	Description
AR_FOR	real	Pre-exponential factor of the forward reaction [Eqn. 85 in Zyvoloski et al. (1997)]. In keeping with the conventional method of defining rate constants, this parameter has units of [(concentration units)^p x s]$^{-1}$, where p is the sum of the exponents on all concentrations in the forward reaction minus 1. Thus, the units of the reaction rate are (concentration units)/s.
EA_FOR	real	Activation energy of the forward reaction (J/mol).
AR_REV	real	Pre-exponential factor of the reverse reaction.
EA_REV	real	Activation energy of the reverse reaction.
STOIC	real	For each solute, the stoichiometric coefficient [the a's in Eqn. 83 in Zyvoloski et al. (1987)] for the particular reaction. If positive, the solute is a reactant; if negative, the solute is a product; and if 0, the solute is not present in the reaction.
RATE_POWER	real	For each solute, the exponent in the rate law [the b's in Eqn. 84 in Zyvoloski et al. (1987)] for the particular reaction. If the corresponding value of STOIC is positive (the solute is a reactant), then b = RATE_POWER; if the corresponding value of STOIC is negative (the solute is a product), then b = - RATE_POWER; and if the corresponding value of STOIC is 0, this value of RATE_POWER is ignored. The reason for this convention is that when the law of microscopic reversibility applies (the exponents in the rate law are the stoiciometric coefficients), the values of STOIC and RATE_POWER are identical. Note also that a value of 0 for RATE_POWER when the corresponding value of STOIC is nonzero implies a zero-th order reaction, and does not mean that the reactant is absent from the reaction.
FL_MULT	real	For each solute, a parameter signifying whether this particular reaction occurs for this solute in the fluid phase. Set the parameter to a nonzero value if the solute in the fluid phase is involved in the reaction, otherwise set the parameter to 0. If this solute is not present in the reaction (the corresponding value of STOIC is 0), then the value of FL_MULT is irrelevant.

Input variable	Format	Description
SB_MULT	real	For each solute, a parameter signifying whether this particular reaction occurs for this solute in the sorbed phase. Set the parameter to a nonzero value if the solute in the sorbed phase is involved in the reaction, otherwise set the parameter to 0. If this solute is not present in the reaction (the corresponding value of STOIC is 0), then the value of SB_MULT is irrelevant.
H_MULT	real	For each Henry's-law species, set this parameter to 1 if the liquid-borne fraction of the solute is reacting, or to -1 if the vapor-borne portion of the species is reacting. For each non-Henry's-law species, this parameter is not used but is read to keep a similar input format to SB_MULT and FL_MULT. Therefore, H_MULT *must* be set for *each* species in *every* reaction if a Henry's-law species is present in the simulation. Both the liquid- and vapor-borne portions of a Henry's-law species cannot react in one reaction. However, *two* reactions, one with the liquid portion reacting and the other with the vapor portion reacting, allow for both phases of the Henry's-law species to react.

The following example, along with the example data input for the **trac** macro, defines the reactive transport problem that is called Run 1 in Section 9.5. The reaction system contains three species (A, B, and C in the description below refer to species 1, 2, and 3, respectively) and the following two chemical reactions.

$$A \Leftrightarrow B$$

$$B \Leftrightarrow C$$

The reactions are reversible, with stoichiometric coefficients of 1, and the powers in the kinetic or equilibrium expressions are all 1 as well. Equilibrium sorption is specified for solute 1 (A) using the isotherm formulations in the **trac** macro. The FL_MULT and SB_MULT input are set so that the chemical reactions here pertain only to solute present in the fluid phase. Group 10 is omitted because for this example there are no Henry's-law species.

The first reaction is kinetically controlled, and forward and reverse rate constants (with no temperature dependence) are given. The second reaction is an equilibrium reaction, and an equilibrium constant (with no temperature dependence) is given. The code will iteratively solve the system in two groups, first with A only, then with B and C coupled. Coupling of the solutes in the equilibrium reaction is necessary for convergence, but solutes that are uncoupled or weakly coupled through "slow" kinetic reactions can be solved separately to minimize computer storage requirements. If the number of finite-element nodes is small enough, all three species could be coupled, and better performance would be expected if a large number of outer iterations are required. To do this in the example above, NGROUP would be set to 1, and a single line of three 1's for GROUP would denote that all three solutes are to be solved simultaneously.

The following is the example of **rxn**

```
rxn
group
2
1            0            0
0            1            1
2            0
kinetic      3.1688e-11   0.          3.1688e-10   0.
equilibrium  1            0.2         0.           1.e-2        1.e-3        1.e-10
1            -1           0
0            1            -1
1            -1           0
0            1            -1
1            1            1
1            1            1
0            0            0
0            0            0
```

6.2.42 Control statement sol (required)

Group 1 - NTT, INTG

Input variable	Format	Description
NTT	integer	Parameter that defines the type of solution required. NTT > 0 - coupled solution. NTT ≤ 0 - heat-transfer-only solution.
INTG	integer	Parameter that defines element integration type. INTG ≤ 0 - Lobatto (node point) quadrature is used, recommended for heat and mass problems without stress. INTG > 0 - Gauss quadrature is used, recommended for problems requiring a stress solution. This is the default for 3-D problems.

The following is an example of **sol**

```
sol
1          -1
```

6.2.43 Control statement solv (Not implemented)

6.2.44 Control statement stea (optional)

No input is associated with this macro statement. The code will attempt to find a steady-state solution if present. Not tested.

This statement enables a 1-D solution in the y-direction (2-D) or z-direction (3-D) when gravity is present to generate an initial steady-state solution.

6.2.45 Control statement stop (required)

No input is associated with this control statement. It signals the end of input, and as such, it always appears as the last line of an input deck.

6.2.46 Control statement strs (Not implemented in this version of FEHM)

6.2.47 Control statement text (optional)

Group 1- WDD1

Input variable	Format	Description
WDD1	character*80	Line of text. A maximum of 80 characters per line are entered. Text is input until a blank line is inserted to signal the end of the control statement. This text is written to the output file (iout).

The following is an example of **text**.

```
text
This is a 2-D model of the PACE problem
It will be used to study thermal effects
user # = -20 to get waste packages
```

6.2.48 Control statement thic (optional)

Input for variable thickness for two-dimensional problems.

Group 1 - JA, JB, JC, THIC (JA, JB, JC - defined on page 19)

Input variable	Format	Description
THIC	real	Thickness of the model domain in the third dimension (m). Default is [1].

The following is an example of **thic**

```
thic
   1          0          0          10.
  -2          0          0           5.
```

In this example, the thickness for all nodes is set to 10 m, after which the nodes defined by zone 2 are set to 5 m. Thus, the thickness is 10 m everywhere except zone 2, where thickness is 5 m.

6.2.49 Control statement time (required)

Time-step and time-of-simulation data.

Group 1 - DAY, TIMS, NSTEP, IPRTOUT, YEAR, MONTH, INITTIME

Group 2 - DIT1, DIT2, DIT3, ITC (as needed)

DAY should be larger than DAYMIN defined in Control Statement **ctrl**. The code proceeds to the next control statement when a blank line is encountered for Group 2. This can be used to generate output at specific times (with multiple group 2s). A contour plot will be drawn at each DIT1 regardless of the input in control statement **cont**. The restart file will be written (or rewritten if one already exists) at each DIT1.

Input variable	Format	Description
DAY	real	Initial time-step size (days).
TIMS	real	Final simulation time (days).
NSTEP	integer	Maximum number of time steps allowed.
IPRTOUT	integer	Print-out interval for nodal information (pressure, enthalpy etc.), as set up under control statement node (i.e., number of time steps).
YEAR	integer	Year that simulation starts.
MONTH	integer	Month that simulation starts.
INITTIME	real	Initial time of simulation (days). For compatibility with older versions, if this parameter is absent, the initial time of simulation will be 0 if no restart file is used or the time in the restart file if one is used.
DIT1	real	Time (days) for time-step change.
DIT2	real	New time-step size (days). If DIT2 < 0, then ABS (DIT2) is the new time-step multiplier.
DIT3	real	Implicitness factor for new time step. DIT3 ≤ 1.0 - backward Euler. DIT3 > 1.0 - second-order implicit scheme.
ITC	integer	New print-out interval.

The following is an example of **time**.

```
time
   30.0        3650.0          20            5          1989           10
    1.0          -1.2          1.0          10
```

6.2.50 Control statement trac

Group 1 - USER_MACRO, ANO, AWC, EPC, UPWGTA

Group 2 - DAYCS, DAYCF, DAYHF, DAYHS

Group 3 - IACCMX, DAYCM, DAYCMM, DAYCMX

Group 4 - NSPECIES

Group 5 - INPUT_MSG

There are two options for group six. If the same diffusion coefficient and dispersivities are to be used for all species of the same type (liquid and/or vapor), then a keyword must be entered. This will make the calculations more efficient and thus should be used if applicable. In the absence of a keyword, the following input is used:

Group 6 - ICNS

Group 7 - IADSF, A1ADSF, A2ADSF, BETADF, DIFFM, TCX, TCY, TCZ

Group 8 - JA, JB, JC, ITRCD (JA, JB, JC - defined on page 19)

Group 9 - HENRY_MODEL, HAWWA(1), HAWWA(2), HAWWA(3), HAWWA(4), HAWWA(5) (only for a Henry's Law species)

Group 10 -JA, JB, JC, ANQO (JA, JB, JC - defined on page 19)

Group 11 - JA, JB, JC, CNSK, T1SK, T2SK (JA, JB, JC - defined on page 19)

Groups 6, 7, 8, 9, 10, and 11 are entered as a unit for each solute. However, for a solid species, only groups 6, 10, and 11 are entered (groups 7, 8, and 9 are not applicable).

Groups 7 and 8 are used to define transport models for which sorption and dispersion parameters are identical. For a liquid or vapor species, only one set of Group 7 parameters should be entered per region. However, for a Henry's-law species, two sets of parameters per region must be entered. For this case, the liquid sorption parameters should be entered on the first line and the vapor sorption parameters on a second line or as a continuation of the first line. Groups 7 and 8 are not applicable to a solid species and should not be entered. Group 7 is read until a blank line is encountered. The model number is incremented by 1 each time a line is read. Group 8 then assigns a transport model number to every node.

For the same diffusion coefficient and dispersivities:

Group 6 - KEYWORD

If only liquid species are present (keyword '*dspl*') or only vapor species are present (keyword '*dspv*'), Group 7 is defined as follows:

Group 7 - DIFFM, TCX, TCY, TCZ

Otherwise, if both liquid and vapor are present (keyword 'dspb'), parameters for both must be entered:

Group 7- DIFFML, TCXL, TCYL, TCZL, DIFFMV, TCXV, TCYV, TCZV

Group 8 - JA, JB, JC, ITRCDSP (JA, JB, JC - defined on page 19)

Group 9 - ICNS

Group 10 -IADSF, A1ADSF, A2ADSF, BETADF

Group 11 -JA, JB, JC, ITRCD (JA, JB, JC - defined on page 19)

Group 12 -HENRY_MODEL, HAWWA(1), HAWWA(2), HAWWA(3), HAWWA(4), HAWWA(5) (only for a Henry's-law species)

Group 13 -JA, JB, JC, ANQO (JA, JB, JC - defined on page 19)

Group 14 - JA, JB, JC, CNSK, T1SK, T2SK (JA, JB, JC - defined on page 19)

Injection nodes must be specified in control statement **flow**.

Input variable	Format	Description
USER_MACRO	character*5	Key word for invoking a solute transport user subroutine. If the word *userc* is placed in this position, then the code invokes a solute-transport user subroutine at each time step. Omit this key word if there is no solute user subroutine for the simulation.
ANO	real	Initial solute concentration set at all nodes for all species unless overwritten by a restart file input or values in group 9 below (moles/kg fluid).
AWC	real	Implicitness factor for solute solution. AWC > 1.0 gives 2nd-order solution. AWC ≤ 1.0 gives 1st-order solution.
EPC	real	Equation tolerance for solute solution.
UPWGTA	real	Upstream weighting term for the solute solution. UPWGTA < 0.5 - UPWGTA is set to 0.5. UPWGTA > 1.0 - UPWGTA is set to 1.0.
DAYCS	real	Time that the solute solution is enabled (days).
DAYCF	real	Time that the solute solution is disabled (days).
DAYHF	real	Time that the flow solution is disabled (days).
DAYHS	real	Time that the flow solution is enabled (days).
IACCMX	integer	Maximum number of iterations allowed in solute solution if time-step multiplier is enabled.
DAYCM	real	Time-step multiplier for solute solution.
DAYCMM	real	Initial time step for solute solution (days).
DAYCMX	real	Maximum time step for solute solution (days).
NSPECIES	integer	Number of solutes simulated.
INPUT_MSG	character*4	Keyword '*ldsp*' specifying longitudinal/transverse dispersion. If x-, y-, z-dispersion is desired, Group 5 is omitted, and dispersivities are input in x, y, and then z order. Otherwise, if longitudinal/transverse dispersion is desired, the keyword '*ldsp'* is entered, and dispersivities are instead input in longitudinal and then transverse order with values for the third dimension omitted.
KEYWORD	character*4	Keyword specifying the same diffusion coefficient and dispersivities are to be used for all species of the same type (liquid and/or vapor). '*dspl'* indicates that only liquid species exist. '*dspv*' indicates that only vapor species exist. *'dspb'* indicates that both liquid and vapor species exist.

DRAFT 3/97

Input variable	Format	Description
ICNS	integer	Phase designation for the ith solute. -2 - Henry's-law species (input and output concentration values are gas concentrations). -1 - Vapor species. 0 - Solid species. 1 - Liquid species. 2 - Henry's-law species (input and output concentration values are liquid concentrations).
IADSF	integer	Adsorption model type for the ith species, ith region. 0 - conservative solute. 1 - linear sorption isotherm. 2 - Freundlich sorption isotherm. 3 - Modified Freundlich sorption isotherm. 4 - Langmuir sorption isotherm.
A1ADSF	real	α_1 parameter in adsorption model.
A2ADSF	real	α_2 parameter in adsorption model.
BETADF	real	β parameter in adsorption model.
DIFFM	real	Molecular diffusion coefficient (m^2/s).
DIFFML	real	Molecular diffusion coefficient for liquid (m^2/s).
DIFFMV	real	Molecular diffusion coefficient for vapor (m^2/s).
TCX	real	Dispersivity in x-direction (m).
TCY	real	Dispersivity in y-direction (m).
TCZ	real	Dispersivity in z-direction (m).
TCXL	real	Dispersivity in x-direction for liquid (m).
TCYL	real	Dispersivity in y-direction for liquid (m).
TCZL	real	Dispersivity in z-direction for liquid (m).
TCXV	real	Dispersivity in x-direction for vapor (m).
TCYV	real	Dispersivity in y-direction for vapor (m).
TCZV	real	Dispersivity in z-direction for vapor (m).
ITRCD	integer	Region number for sorption and dispersion parameters given in group 7 (no keyword) or for sorption parameters given in group 11. Default is [1].
ITRCDSP	integer	Region number for dispersion parameters given in group 7 (keyword). Default is [1].

Input variable	Format	Description
HENRY_MODEL	integer	Flag denoting which model to be used for defining the temperature dependence of the Henry's-law constant. 1 - van't Hoff model. 2 - multiparameter fit to experimental data (used for carbonate system).
HAWWA(1)	real	Term in Henry's-law temperature-dependence model. For model 1 - parameter value is A_H. For model 2 - parameter value is $A_{H,1}$.
HAWWA(2)	real	Term in Henry's-law temperature-dependence model. For model 1 - parameter value is ΔH_H. For model 2 - parameter value is $A_{H,2}$.
HAWWA(3)	real	Term in Henry's-law temperature-dependence model. For model 1 - not used. For model 2 - parameter value is $A_{H,3}$.
HAWWA(4)	real	Term in Henry's-law temperature-dependence model. For model 1 - not used. For model 2 - parameter value is $A_{H,4}$.
HAWWA(5)	real	Term in Henry's-law temperature-dependence model. For model 1 - not used. For model 2 - parameter value is $A_{H,5}$.
ANQO	real	Initial concentration of tracer, which will supercede the value given by ANO in group 1. Note that if initial values are read from a restart file, these values will be overwritten. Units are moles per kg vapor or liquid for a liquid, vapor, or Henry's-law species and moles per kg of solid for a solid species. Default is [0].
CNSK	real	Injection concentration at inlet node (moles per kg liquid or vapor). If fluid is exiting at a node, then the in-place concentration is used. If CNSK < 0, then the concentration at that particular node will be held at a concentration of abs(CNSK). Default is [0] for all unassigned nodes.
T1SK	real	Time (days) when tracer injection begins. Default is [0].
T2SK	real	Time (days) when tracer injection ends. Default is [0].

In the following example of **trac**, three liquid solutes are simulated. The solute transport solution is turned on as the heat and mass solution is turned off at day 3.6525e6. Solute 1 sorbs with an equilibrium sorption K_d value of 0.1; solutes 2 and 3 exhibit no sorption. All three solutes have the same transport parameters, although this is not a requirement of the code, even when the solutes are coupled through chemical reactions. All solutes start at a concentration of 0 within the model (in this case, a one-dimensional column). Solute 1 is injected at a concentration of 1 for a short time interval; there is no source for solutes 2 or 3. This example is meant to be used in combination with the example given for the **rxn** macro, which defines a system of chemical reactions among the three solutes. Therefore, solutes 2 and 3 are generated only through chemical reactions (neither of these solutes appear in the system initially, and there is no injection source term for either solute). The second example of **trac** is modified for use with longitudinal and transverse dispersion. The third and fourth examples are the above two examples of **trac**, modified to illustrate keyword use for the same diffusion coefficient and dispersivities. The reactive transport problem specified here and in the example **rxn** macro is discussed further in Section 9.5.

```
trac
0          1          1.e-6      .5
3.6525e6   5.47875e6  3.6525e6   5.47875e6
50         1.2        3.6525e3   3.6525e4
3
1
1          0.1        0.         1.         1.e-10     50         1.e-30   1.e-30

   1          0          0          1

   1          0          0          0.

   1          202        201        1.         3.6525e6   3.689025e6

   1
   0          0          0          1.         1.e-10     50         1.e-30   1.e-30

   1          0          0          1

   1          0          0          0.


   1
   0          0          0          1.         1.e-10     .033333    1.e-30   1.e-30

   1          0          0          1

   1          0          0          0.

```

```
trac
   0          1          1.e-6      .5
   3.6525e6   5.47875e6  3.6525e6   5.47875e6
   50         1.2        3.6525e3   3.6525e4
   3
ldsp
   1
   1          0.1        0.         1.         1.e-10     50         5.0

   1          0          0          1

   1          0          0          0.

   1          202        201        1.         3.6525e6   3.689025e6

   1
   0          0          0          1.         1.e-10     50         5.0

   1          0          0          1

   1          0          0          0.


   1
   0          0          0          1.        1.e-10     .0333       .00333

   1          0          0          1

   1          0          0          0.


```

```
 trac
   0          1          1.e-6       .5
 3.6525e6   5.47875e6  3.6525e6   5.47875e6
  50        1.2        3.6525e3   3.6525e4
   3
 dspl
  1.e-10     .033333   1.e-30     1.e-30

  -1          0          0          1

   1
   1          0.1        0.         1.

   1          0          0          1

   1          0          0          0.

   1        202        201          1.         3.6525e6   3.689025e6

   1
   0          0          0          1.                    .

   1          0          0          1

   1          0          0          0.


   1
   0          0          0          1.

   1          0          0          1

   1          0          0          0.


```

```
trac
   0          1            1.e-6        .5
 3.6525e6   5.47875e6    3.6525e6     5.47875e6
  50          1.2        3.6525e3     3.6525e4
   3
ldsp
dspl
 1.e-10      .0333        .00333

  -1          0            0            1

   1
   1          0.1          0.           1.

   1          0            0            1

   1          0            0            0.

   1        202          201            1.          3.6525e6    3.689025e6

   1
   0          0            0            1.                      .

   1          0            0            1

   1          0            0            0.


   1
   0          0            0            1.

   1          0            0            1

   1          0            0            0.


```

6.2.51 Control statement user (optional)

Group 1 - KK

Input variable	Format	Description
KK	integer	Integer number passed to subroutine **user** for user-defined input parameters. This user subroutine call differs from the one invoked in the control file in that whereas that subroutine is called at every time step, this one is called only at the beginning of the simulation to set parameters that do not change later in the simulation.

6.2.52 Control statement vcon (optional)

Group 1 - IVCON(i), VC1F(i), VC2F(i), VC3F(i)

Group 2 - JA, JB, JC, IVCND (JA, JB, JC - defined on page 19)

The parameter (i) is incremented each time Group 1 is read. Group 2 lines will refer to this parameter. Group 1 is ended with a blank line.

Input variable	Format	Description
IVCON(i)	integer	Model type for ith conductivity model. IVCON(i) = 1 - linear variation of thermal conductivity with temperature. IVCON(i) = 2 - square-root variation of thermal conductivity with liquid saturation.
VC1F(i)	real	Reference temperature (^oC) for IVCON(i) = 1. Conductivity $\left(\frac{W}{m \cdot K}\right)$ at liquid saturation = 1 for IVCON(i) = 2.
VC2F(i)	real	Reference conductivity $\left(\frac{W}{m \cdot K}\right)$ for IVCON(i) = 1. Conductivity $\left(\frac{W}{m \cdot K}\right)$ at liquid saturation = 0 for IVCON(i) = 2.
VC3F(i)	real	Change in conductivity with respect to temperature for IVCON(i) = 1. Not used for IVCON(i) = 2.
IVCND	integer	Number referring to the sequence of models read in Group 1. The default is [1].

6.2.53 Control statement velo (optional)

The input for this macro is identical to macro **flxo**, except that velocities instead of fluxes are calculated (see page 36).

6.2.54 Control statement wlbr (optional)

Not supported in this implementation of FEHM.

6.2.55 Control statement zone (optional)

Geometric definition of grid for input parameter assignment. The default is input by nodes.

Group 1- IZONE

Group 2- X1, X2, X3, X4 (for 2-D) or X1, X2, X3, X4, X5, X6, X7, X8 (for 3-D)

Group 3- Y1, Y2, Y3, Y4 (for 2-D) or Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8 (for 3-D)

Group 4- Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8 (for 3-D problems only)

The following alternate form of input may be used (starting with Group 2):

Group 2 - MACRO

Group 3 - XG, YG (for 2D) or XG, YG, ZG (for 3-D) [used with 'list' option]

or

Group 3 - NIN, NCORD(1), . . . , NCORD(NIN) [used with 'nnum' option]

The geometric-zone description is implemented by defining geometric regions. The coordinates given in Groups 2, 3, and 4 refer to the node positions shown in Fig. 2. All properties defined by node (JA, JB, JC) in any control statements may be defined by **zone**. In the previous macro descriptions, if JA < 0, then the zone IZONE = ABS (JA) is referenced.

It is a good policy to refer to the input check file to insure that node assignments have been made as expected. When x-, y-, z-coordinates are used to define zones, boundaries of those zones may be slightly different than specified. This effect is due to the inclusion of volume from elements adjoining included nodes.

When macro statements **dpdp** and **dual** are used, additional zone definitions are automatically generated. These are identified by zones 101-200 for the first set of matrix nodes and 201-300 for the second set of matrix nodes. For example, Zone 101 corresponds to the matrix material occupying the same space occupied by the fracture identified by Zone 1. Furthermore, Zone 201 refers to the second matrix layer in the **dual** control statement.

The macro **zone** must precede the usage of a ZONE reference. **zone** is ended with a blank line. **zone** can be called more than once and regions redefined. When this is done, all previous zone definitions are eliminated. A node may be included in only a single zone at a time.

Input variable	Format	Description
IZONE	integer	Zone identification number for geometric input.
X1-X8	real	X-coordinates defining zone IZONE.
Y1-Y8	real	Y-coordinates defining zone IZONE.
Z1-Z8	real	Z-coordinates defining zone IZONE.
MACRO	character*4	String denoting alternate input format. MACRO = "list" - read a list of x-, y-, z-coordinates, one set per line until a blank line is encountered. The nodes corresponding to these coordinates make up the zone. MACRO = "nnum" - read the number of specified nodes, followed by the node numbers. These comprise the zone.
XG	real	X-coordinate of node to be included in IZONE.
YG	real	Y-coordinate of node to be included in IZONE.
ZG	real	Z-coordinate of node to be included in IZONE.
NIN	integer	Number of nodes in IZONE.
NCORD(i)	integer	NIN node numbers of the nodes to be included in IZONE.

The following is an example of **zone**

```
zone
   1
   0.00        1000.00      1000.00        0.00
1075.00        1074.00      1079.00     1080.00
   2
   0.000       1000.000     1000.00        0.00
 870.000        869.000     1074.00     1075.00
   3
   0.000       1000.000     1000.00        0.000
 860.000        859.000      869.000     870.000
   4
   0.000       1000.000     1000.00        0.000
 780.000        779.000      859.000     860.000
  5
   0.000       10000.00     1000.00        0.000
 730.000        730.000      779.000     780.000
   6
   0.000       1000.000     1000.00        0.000
 700.000        700.000      730.000     730.000
```